PLATAFORMA DIDÁTICA PARA TREINAMENTO EM AUTOMAÇÃO DE SUBESTAÇÕES UTILIZANDO SISTEMAS EMBARCADOS.

DANIELY DA S. MACHADO, MARIANA RODRIGUES, RODRIGO M. A. ALMEIDA

IESTI - Universidade Federal de Itajubá
CEP 37500-903, Itajubá, Minas Gerais, Brasil

E-mails: daniely.dsm@gmail.com, rodrigues.mariana@gmail.com, rodrigomax@unifei.edu.br

Abstract — It is presented in this document a student/operator training set for substation automation, composed by a hardware system which simulates substation components and a core which implements maneuver control rules. Electrical components are simulated by independent relay circuit boards, while the control core includes a microcontroller programmed in C language and the electronic interface necessary to trigger the circuit boards. The usage of independent circuit boards allows the assembly of several substation architectures, and the software model is built so that its expansion is easy and accessible, taking into consideration constructive and security aspects. Tests show that all interlocking and security rules are followed, making the set a viable alternative for student training.

Keywords — Substation Automation, Embedded Systems, Training Set.

Resumo — Neste trabalho, é apresentado um sistema para treinamento de estudantes ou operadores em automação de subestações, composto por um sistema de hardware para simulação dos componentes da subestação e um núcleo para implementação das regras de controle que realizam as manobras. Os componentes elétricos são simulados por meio de placas autônomas com relés, enquanto o núcleo de controle contém um microcontrolador programado em linguagem C e a interface necessária para o acionamento das placas eletrônicas. O uso de placas autônomas permite montar várias topologias de subestação, e o software é construído visando a facilidade de uso e escalabilidade, levando em consideração aspectos construtivos e de segurança. Os testes demonstram conformidade com as regras de intertravamento e de segurança para o modelo de subestação apresentado, tornando o conjunto uma alternativa viável para o treinamento de alunos.

Palavras-chave — Automação de subestações, Sistemas Embarcados, Kit Didático.

1 Introdução

Subestações são parte importante do sistema elétrico, sendo utilizadas para direcionar o fluxo de potência no sistema elétrico. Podem realizar a interface de uma unidade geradora ao sistema, a conversão entre faixas de tensão ou a interligação de várias linhas de fornecimento que se encontram em um mesmo nível de tensão (GILES, 1970).

A automação de subestações aumenta a confiabilidade e a flexibilidade do sistema, diminui o custo operacional e ainda pode auxiliar na identificação e recuperação de falhas (BRAND et al., 2003). Não por acaso, as concessionárias de energia caminham hoje para a automação de seus sistemas de distribuição, os quais passam a ser controlados por centros de operação remotos.

Para orientar o fluxo de energia, uma subestação realiza manobras com seus dispositivos de chaveamento. Manobras são seqüências de comandos de chaveamento com um objetivo, como por exemplo ligar uma cidade a um ponto da rede de distribuição. Cada passo da manobra é realizado de forma independente, e um próximo passo só é executado se todos os passos anteriores não apresentaram erros (BRAND et al., 2003).

Os dispositivos de chaveamento, como chaves seccionadoras e disjuntores, permitem as operações das subestações. Chaves seccionadoras auxiliam o isolamento de partes da rede elétrica, e podem ou não serem abertas sob carga. Os disjuntores são responsáveis pela interrupção do circuito propriamente dita, e devem ser capazes de extinguir qualquer arco voltaico gerado por ela (ROSSI, 2004).

O acionamento dos dispositivos de chaveamento é parte crítica da automação de uma subestação devido ao risco às pessoas próximas do equipamento no momento do chaveamento. Para verificar o seu estado de chaveamento, os dispositivos possuem contatos auxiliares acoplados aos principais, cujas posições representam o estado de chaveamento principal (aberto ou fechado). A informação dos contatos auxiliares pode ser transformada em uma informação binária (verdadeiro ou falso) com o auxílio de circuitos apropriados (BRAND et al., 2003).

Em subestações de alta e média tensão, as chaves seccionadoras só podem ser acionadas a vazio (sem carga). Assim, para garantir a segurança do operador e dos equipamentos, o controle das chaves deve seguir regras de intertravamento na sua operação. O intertravamento deve garantir que apenas manobras corretas sejam executadas, evitando a

ocorrência de erros operacionais (GALDEANO, [s.d.]).

Antes da manobra ser iniciada, as regras de intertravamento devem ser verificadas e, caso algum erro ou algum impedimento para a manobra seja detectado, a manobra é abortada. É possível detectar erros de acionamento contrapondo o estado esperado e o estado real do dispositivo, este obtido pelos seus contatos auxiliares (BRAND et al., 2003).

Neste trabalho, é apresentada uma abordagem de ensino e simulação de conceitos de automação de subestações baseada em uma configuração flexível e sistemas embarcados, utilizando programação em linguagem C em microcontroladores. A abordagem permite o aluno/operador aprender os conceitos de automação de subestações de maneira ampla (pelo sistema flexível) e com segurança (já que nenhum elemento de alta tensão está envolvido).

Optou-se por uma abordagem mista de hardware e software para permitir que o usuário possa simular falhas no sistema de sistema físico. Isto traz uma resposta mais parecida com a real, onde os problemas de *bouncing* e tempo de acionamento das chaves pode ser simulado por relés e contatores.

2 Apresentação do conjunto

O conjunto proposto é constituído de duas partes: uma implementada em hardware e uma em software. O hardware é responsável por simular o funcionamento de uma chave real, com todos os problemas intrínsecos a este tipo de acionamento como *bouncing*, atrasos na resposta e modos de falha. O software é responsável por permitir o controle e a automação destes componentes utilizando um modelo de software que pode ser modificado pelo usuário.

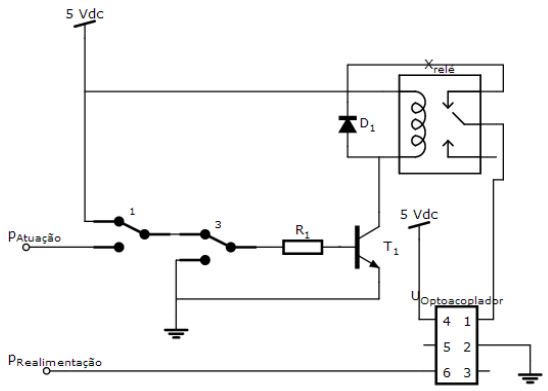

Figura 1 –Circuito de interface projetado com possibilidade de simulação de falhas

Os componentes da subestação são simulados utilizando placas autônomas de circuitos eletrônicos com relés, os quais serão acionados pelo controlador através de um transistor, conforme Figura 1. O estado de chaveamento é disponibilizado para o controlador por meio de um optoacoplador. Estes circuitos também possuem chaves que possibilitam mantê-lo num estado definido: ligado ou desligado. Desta forma, é possível simular falhas em cada um dos componentes da subestação.

A utilização de placas autônomas permite a construção de modelos em qualquer topologia de subestação, possibilitando o treinamento de várias situações e configurações com o mesmo conjunto de hardware, além de inserir no sistema de treinamento situações reais como atraso na resposta do acionamento e *bouncing* das chaves.

O núcleo de controle contém um microcontrolador e a interface eletrônica de entrada/saída necessária para o acionamento das placas eletrônicas. O microcontrolador contém um software de controle das manobras e acionamentos, além de rotinas para a detecção de erros nos estados das chaves.

O modelo do software foi desenvolvido em linguagem C. Esta é uma linguagem de programação amplamente utilizada em sistemas embarcados devido à sua característica "baixo nível", significando que o programador possui um bom controle do hardware sem perder as vantagens de linguagens ditas de "alto nível", mais amigáveis na programação (BARR, 1999).

A linguagem C ainda torna a aplicação portátil entre diferentes dispositivos, já que a programação se torna independente do microprocessador utilizado (BARR, 1999). Isso aumenta ainda mais a flexibilidade do sistema, já que qualquer microcontrolador pode ser utilizado para o circuito de controle.

Optou-se neste trabalho por construir o sistema em camadas crescentes em complexidade. Deste modo as funções similares estão agrupadas e organizadas num mesmo nível. O modelo desenvolvido é apresentado na Figura 2.

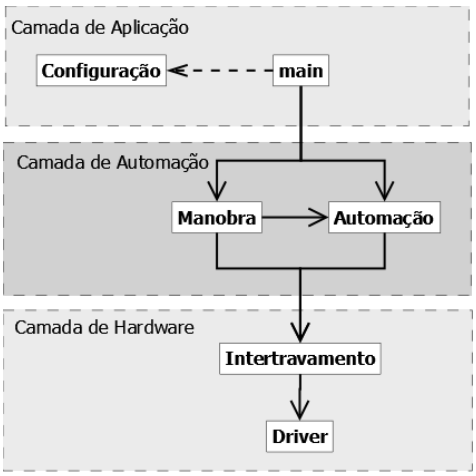

Figura 2 - Modelo de software

Cada camada possui uma responsabilidade específica, seja o desenvolvimento da aplicação, a automatização dos processos de checagem ou a manutenção da segurança. Estas camadas são:

Camada de aplicação: é a interface do programador com o restante do sistema, podendo fornecer informações sobre quais manobras foram executadas, e

realizar manobras manualmente (por meio de botões de acionamento).

Camada de automação: a biblioteca *Manobra* implementa as manobras disponíveis e por isso é dependente da topologia da subestação. Já a biblioteca *Automação* armazena manobras auxiliares para recuperação do sistema em caso de falha. Nesta biblioteca, o programador define quais são as manobras auxiliares e qual a prioridade de cada uma delas.

Camada de Hardware: realiza a interface direta com o acionamento dos componentes. A biblioteca *Driver* é responsável por ligar/desligar os comutadores, e a biblioteca *Intertravamento* garante que as regras de intertravamento dos comutadores sejam obedecidas.

A organização do software em camadas auxilia o tratamento e depuração do software, além de facilitar sua expansão caso o número de entradas ou saídas aumente.

3 Desenvolvimento do Software

Foi utilizado como base uma planta de simulação de subestação existente no Laboratório de Sistemas de Automação (LSA) do Instituto de Engenharia de Sistemas e Tecnologia da Informação (IESTI) da Universidade Federal de Itajubá (UNIFEI). Esta é representada pelo diagrama unifilar da Figura 3. A planta conta com 3 linhas elétricas, 10 chaves seccionadoras e três disjuntores.

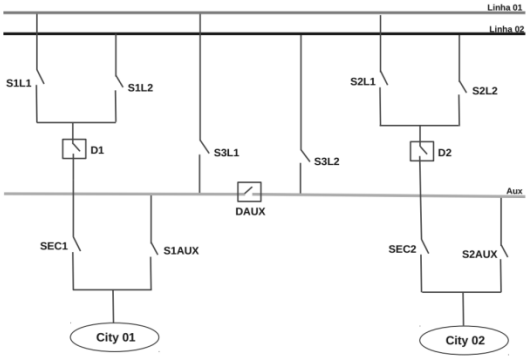

Figura 3 - Diagrama unifilar da maquete

Um dos motivos para a utilização desta topologia é aumentar a capacidade do laboratório, permitindo que os alunos tenham duas plantas de simulação com topologias iguais, mesmo que com modos de acionamentos diferentes.

3.1 Manobras do Sistema

O conjunto de manobras possíveis na planta apresentada é reduzido porém suficiente para as experiências em sala de aula, além de permitir aos alunos um primeiro contato com manobras e operações em uma subestação.

Como as chaves seccionadoras só podem ser acionadas a vazio (GALDEANO, [s.d.]), cada manobra para ligar ou desligar uma cidade na planta deve incluir um disjuntor. Partindo deste princípio, as manobras do sistema são exibidas pela Tabela 1, que explicita quais componentes devem estar fechados em cada uma.

Tabela 1 - Manobras disponíveis na planta utilizada

	City 01	City 02
Linha 01	Manobra 0 . S1L1 . D1 . SEC1	Manobra 3 . S2L1 . D2 . SEC2
Linha 02	Manobra 1 . S1L2 . D1 . SEC1	Manobra 4 . S2L2 . D2 . SEC2
Aux	Manobra 2 . S3L2 . DAUX . S1AUX	Manobra 5 . S3L1 . DAUX . S2AUX

3.2 Regras de intertravamento

Com o intuito de simular o ambiente real, algumas regras de distribuição foram desenvolvidas de forma a restringir a gama de operações na planta:

- A linha auxiliar pode alimentar apenas uma cidade por vez;
- Cada cidade é alimentada por apenas uma linha;
- As linhas principais (Linha 01 e Linha 02) podem alimentar mais de uma cidade, caso a oferta de potência supra a demanda.

As duas primeiras regras são estáticas e podem ser inferidas apenas com o estado dos elementos da subestação, ligados ou desligados.

A terceira regra necessita de uma medição de uma grandeza contínua, em geral corrente, pelo barramento, e ainda não foi implementada nesta primeira versão do Software.

Assim, a partir das manobras e das regras de distribuição, podem ser inferidas as regras de intertravamento da planta:

- As chaves S1L1 e S1L2 são mutuamente excludentes, ou seja, uma só pode ser acionada se a outra estiver desligada, e os dois componentes nunca entram em operação paralelamente (A cidade é alimentada por apenas uma linha);

- As chaves S2L1 e S2L2 são mutuamente excludentes (A cidade é alimentada por apenas uma linha);
- As chaves S1AUX e S3L1 são mutuamente excludentes (Deve sempre haver um disjuntor na manobra);
- As chaves S2AUX e S3L2 são mutuamente excludentes (Deve sempre haver um disjuntor na manobra);
- As chaves S1AUX e S2AUX são mutuamente excludentes (A linha auxiliar pode alimentar apenas uma cidade por vez).
- O disjuntor D1 só pode ser ligado caso o par <S1L1, SEC1 > ou o par <S1L2, SEC1 > já estiver ligado (Manobra 0 e Manobra 1);
- O disjuntor D2 só pode ser ligado se ou o par <S2L1, SEC2 > ou o par <S2L2, SEC2 > já estiver ligado (Manobra 3 e Manobra 4).
- O disjuntor DAUX só pode ser ligado quando ou o par <S1AUX, S3L2 > ou o par <S2AUX, S3L1 > já estiver ligado (Manobra 2 e Manobra 5).

3.3 Desenvolvimento das bibliotecas

Na biblioteca *Driver*, foram desenvolvidas três funções para cada comutador: *On*, *Off* e *Test*, as quais, respectivamente, ligam, desligam e testam o estado de chaveamento de cada comutador. As funções enviam o comando para o comutador e aguardam o sinal de retorno através da realimentação realizada pelos contatos do relé, simulando um contato real. Apenas quando o sinal correto chega ao controlador o mapa interno de variáveis é alterado, indicando que a operação foi realizada corretamente. No caso de algum problema, o sistema aguarda um tempo máximo antes de remover o comando e sinalizar que há falhas na operação. O tempo máximo esperado também ajuda o controlador a lidar com o aspecto mecânico dos comutadores, evitando assim problemas de *bouncing*. Há ainda na biblioteca uma rotina de inicialização que configura as entradas e saídas necessárias para os acionamentos.

Na biblioteca *Intertravamento*, foram desenvolvidas rotinas que testam as regras de intertravamento antes do chaveamento de qualquer um dos componentes. As rotinas testam as condições e, não havendo impedimento, utilizam as rotinas da biblioteca *Driver* para realizar o chaveamento dos comutadores. Além de uma rotina de inicialização, existe também a rotina *ProcuraErro()*, a qual verifica inconsistências entre os comandos emitidos e os estados das chaves, e retorna qual chave apresenta inconsistência.

A biblioteca *Manobra* contém rotinas que ligam e desligam as manobras disponíveis para a planta em questão (Tabela 1). Para acionar os comutadores, são utilizadas as rotinas da biblioteca *Intertravamento*, o que garante que os chaveamentos sejam feitos com segurança. Para facilitar a operação de manobras pelo usuário da biblioteca, foram criadas funções genéricas para ligar, desligar ou testar manobras, cujo parâmetro corresponde ao número da manobra, conforme a Tabela 1. As funções genéricas utilizam vetores de função para realizar a chamada da manobra certa. A função específica chamada pela função genérica depende do índice fornecido como parâmetro. As funções de operação das manobras verificam as condições iniciais para que a manobra possa ser realizada, e ligam os componentes um a um (lembrando que, pelo fato da chave seccionadora ter que operar a vazio, o disjuntor é o último componente a ser ligado). Cada passo da manobra é verificado, e caso um comando tenha sido emitido e resultado em erro, a execução da manobra é cancelada e todas as ações realizadas naquela manobra são desfeitas.

Por fim, a biblioteca *Automação* contém uma única rotina, *TesteAutomacao()*, que procura falhas no sistema e age para consertá-las. A rotina é auxiliada por duas matrizes: uma que armazena a relação entre os componentes do sistema e as manobras disponíveis, e outra que armazena as manobras alternativas para o caso de problemas em cada uma das manobras. Caso a função *ProcuraErro()* da biblioteca *Intertravamento* encontre alguma inconsistência, é verificada qual manobra está ligada e contém o dispositivo com problema. Quando esta manobra é encontrada, ela é desligada e são tentadas as manobras alternativas. As manobras alternativas são cadastradas em uma matriz, cujas linhas correspondem ao número da manobra conforme a Tabela 01 e as colunas correspondem ao número de uma manobra alternativa.

3.4 Considerações sobre atomicidade

É característica dos microcontroladores tratar interrupções. Interrupções são sinais que interrompem o programa principal para executar um outro código, chamado de rotina de interrupção. Em grande parte dos casos, as interrupções auxiliam a supervisão de periféricos do microcontrolador, já que enquanto o periférico não termina a tarefa atual, o processador está livre para executar outras tarefas, aumentando então a eficiência do sistema (BARR, 1999). As interrupções podem ser habilitadas ou desabilitadas, e seu número e uso variam conforme o microcontrolador utilizado (CATSOULIS, 2005).

Uma instrução em linguagem C quando compilada pode se transformar em vários comandos em linguagem Assembly — a linguagem de programação que o microcontrolador entende. Assim, o microcontrolador pode ser interrompido no meio de uma instrução na linguagem C. Em algumas aplicações, existem ações consideradas críticas ou atômicas, ou seja, que não podem ser interrompidas (SIMON, 2005).

No conjunto proposto, as funções de LIGA e DESLIGA das chaves não podem ser interrompidas, correndo o risco de gerar inconsistência no estado das chaves. Tal problema pode resultar em um acionamento errado de uma chave, o que poderia causar um curto-circuito.

Assim, as funções de LIGA e DESLIGA das chaves são tratadas como funções atômicas. Para atingir este resultado, as interrupções do microcontrolador são desabilitadas antes de alterar o estado da chave, sendo habilitadas depois que a tarefa é executada. O processo é ilustrado pela Figura 4.

Figura 4- Fluxograma das funções de controle dos comutadores.

4 Testes e Resultados

Nesta fase de desenvolvimento e testes, o sistema embarcado foi substituído pelo kit didático NEO201 (EXSTO, 2014), baseado no microcontrolador PIC18F4550 (MICROCHIP, 2014). O kit possui, dentre outros periféricos, um LCD Alfanumérico (16x2 posições), um barramento de LED's e um teclado de 16 teclas, conforme mostrado pela Figura 5.

Para realizar o teste das bibliotecas, foi criado um software principal que utiliza o teclado e o display de LCD de uma placa de desenvolvimento microcontrolada para simular o comando de manobras ou falhas e fornecer ao operador o feedback da operação. O barramento de LED's disponível permitiu conferir se as saídas estavam sendo acionadas corretamente. O programa supervisiona o teclado e, dependendo da tecla pressionada, liga ou desliga uma manobra.

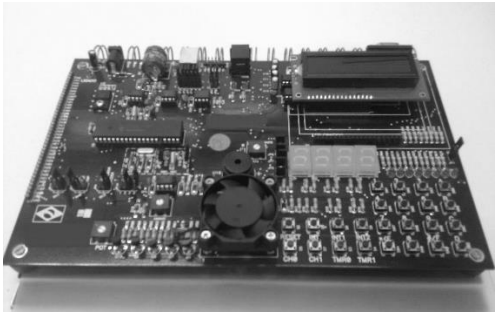

Figura 5 - Placa de desenvolvimento utilizada nos testes.

O barramento de LED's permitiu a conferência da ordem de acionamento dos comutadores, e pode ser comprovado que as chaves seccionadoras são acionadas a vazio e o disjuntor é chaveado por último. Os *delays* inseridos para *debouncing* permitem a confirmação visual dos resultados.

Foi realizado um teste exaustivo para teste do intertravamento das operações e sua conformidade com as condições estabelecidas na seção 3.2. O resultado destes testes é resumido pela Tabela 2, a qual indica quais manobras nas colunas puderam ser ligadas simultaneamente quando a manobra da linha estivesse ligada. Os números das manobras são os mesmos da Tabela 1.

Tabela 2 - Manobras simultâneas

Manobras	0	1	2	3	4	5
0		N	N	S	S	S
1	N		N	S	S	S
2	N	N		S	S	N
3	S	S	S		N	N
4	S	S	S	N		N
5	S	S	N	N	N	

Percebe-se pela Tabela 2 que as regras de intertravamento apresentadas foram respeitadas.

Também foram realizados testes para simulação de falhas. O acionamento de um botão foi configurado para alterar o estado de um dos elementos, por exemplo o disjuntor D1 do esquema da Figura 02. Se alguma das manobras que utiliza o disjuntor D1 estivesse ligada (Manobra 0 ou Manobra 1), a rotina de automação encontra a inconsistência e automaticamente desliga a manobra em uso e utiliza a manobra alternativa para continuar a alimentação. A Figura 6 apresenta o sistema detectando uma falha na chave S1L1 e automaticamente trocando a alimentação da cidade 1 para o barramento auxiliar, representado pelos três LED's acessos.

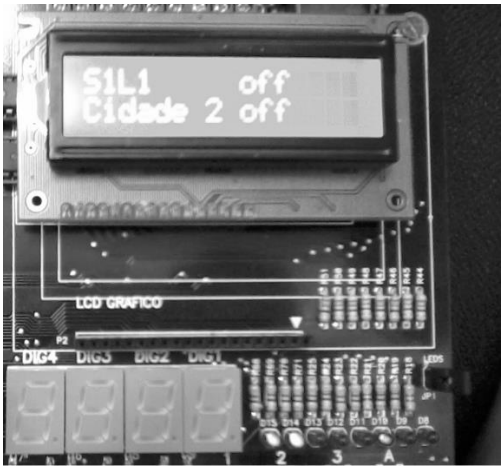

Figura 6 - Detecção de falhas e recuperação automática do sistema.

5 Conclusão

O modelo de software proposto permite cumprir de modo simplificado as regras rotineiras em uma subestação, podendo ser utilizado para treinamento na automação de subestações. A construção do sistema em camadas permite a fácil expansão do sistema, possibilitando que os alunos possam testar novas alternativas e topologias modificando apenas a camada de manobras, o que não é possível com a planta instalada no laboratório.

A realização de testes de funcionamento de outras abordagens de automação também é simplificada, já que todas as medidas de segurança estão implementadas em camadas não acessíveis ao usuário. Isto, aliado às vantagens da programação em linguagem C, permitem ao aluno desenvolver sistemas bastante complexos do ponto de vista computacional.

Os testes realizados com a plataforma desenvolvida são mais eficientes a medida que o hardware desenvolvido consegue simular com maior fidelidade os eventos reais: *bouncing* das chaves, atraso na resposta e falhas nos elementos.

Por fim, a flexibilidade do sistema, atrelado ao seu baixo custo e um bom nível de detalhes simulados, torna o conjunto apresentado uma alternativa viável para o treinamento dos alunos na automação de subestações, bem como na prática de desenvolvimento para sistemas embarcados.

As próximas possibilidades de aprimoramento incluem a criação de uma interface com um computador (por comunicação serial) para aprimorar a interação do usuário com o sistema e permitir a simulação no ambiente virtual de elementos inviáveis de serem adicionados na planta. Com esta funcionalidade, é possível mostrar ao usuário de forma mais intuitiva o estado do sistema ou possíveis falhas que ocorram.

Outra possibilidade de aprimoramento é a adaptação do sistema para uma forma mais genérica, na qual o usuário não necessite construir rotinas para cada componente ou manobra, utilizando-se de estruturas de dados mais robustas.

Referências Bibliográficas

BARR, Michael. Programming Embedded Systems in C and C++. Estados Unidos: O'Reilly& Associates, Inc, 1999. 191p.

BRAND, Klaus-Peter; LOHMANN, Volker; WIMMER, Wolfgang Substation automation handbook. Suíça: Utility Automation Consulting Lohmann, 2003. 397p.

CATSOULIS, John. Designing Embedded Hardware. Estados Unidos: O'Reilly & Associates, Inc, 2005. 398p.

EXSTO Tecnologia. XM111 - Banco de Ensaios para Microcontroladores PIC18F. http://www.exsto.com.br/produto-microcontroladores-e-microprocessadores-xm111-banco-de-ensaios-para-microcontroladores-pic18f-desenvolvimento Acessado em (10/01/2014).

GALDEANO, DuilioManzano. Comando e controle de subestações: E912.Itajubá: [s.n.], [s.d.]. 100p.

GILES, Robert L. Layout of E.H.V. substations. London: Cambridge University, 1970. 223 p. (IEE Monograph Series 5).

MICROCHIP Technology Inc. PIC18F4550. http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010300 Acessado em (10/01/2014).

ROSSI, Ronaldo. Subestações Elétricas de Alta Tensão - Operação e Manutenção. Itajubá/MG: [s.n.], 2004. 164p.

SIMON, David E. An Embedded Software Primer. Índia: Pearson Education, 2005. 424p.